MODEL BASED LONGITUDINAL CONTROL OF HEAVY DUTY VEHICLES

A Thesis

by

VAMSI KRISHNA VEGAMOOR

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Chair of Committee,	Swaroop Darbha
Committee Members,	Swaminathan Gopalswamy
	Alireza Talebpour
Head of Department,	Andreas A. Polycarpou

December 2018

Major Subject: Mechanical Engineering

ABSTRACT

Model based design has proven to be an efficient approach for developing and testing embedded systems. In this work, we attempt to apply this approach to heavy duty vehicles with the goal of implementing a longitudinal controller for velocity tracking. A model was developed in Dymola and parameter identification was performed using specifically designed experiments with a tractor – trailer. The core of longitudinal dynamics of the model involves an engine torque map generated from experimental data. Based on this model, a PID controller was designed and tuned using a closed loop desktop simulation with the plant model. Finally, the controller was implemented in field tests and its performance was verified. We show that this modeling and controller development process can be completed by utilizing the onboard SAE J1939 CAN bus, without the need for any manufacturer privileged information.

The model-based controller developed was found to be stable and was able to track a wide range of velocities to within 0.5 m/s (~ 1 MPH) of the desired value. Moreover, the plant model developed in Dymola was confirmed to have sufficient fidelity to be reliably be used for any new control algorithm development in the future.

ACKNOWLEDGEMENTS

I would like to thank my committee chair, Dr. Swaroop Darbha, and my committee members, Dr. Swaminathan Gopalswamy and Dr. Alireza Talebpour for their guidance throughout the course of this research. I would also like to extend my gratitude to Dr. Sivakumar Rathinam for his support.

I am also grateful to Mr. Mike Lukuc and Texas A&M Transportation Institute for supporting this research project and allowing access to their testing facilities at the RELLIS campus.

Finally, I am thankful to my parents for supporting me through all my endeavors.

CONTRIBUTORS AND FUNDING SOURCES

This work was supported by a dissertation committee consisting of Dr. Swaroop Darbha and Dr. Swaminathan Gopalswamy of the Department of Mechanical Engineering and Dr. Alireza Talebpour of the Department of Civil Engineering.

All work for the dissertation was completed independently by the student. Graduate study was supported through research and teaching assistantships from Texas A&M University.

TABLE OF CONTENTS

LIST OF FIGURES

Page

LIST OF TABLES

Page

1. INTRODUCTION

Development of reliable longitudinal control algorithms is often the first step in enabling autonomy for heavy duty vehicles. Model based control has been widely accepted to yield robust performance in cases where the plant model is invertible. While a significant amount of work has been done in model development and control for tractor-trailer vehicles, most researchers have either depended on the availability of detailed specifications of components from the manufacturer or relied on aftermarket modifications. For example, Lu and Hedrick [1] present a culmination of work, which involves detailed modeling of the turbocharged diesel engine using fuel maps. Such maps are usually privileged information and not available in published literature. In this work, we propose a method to develop a plant model for a heavy-duty vehicle using data collected from onboard SAE J1939 CAN bus, without relying on specialized probes or availability of confidential information from OEMs (Original Equipment Manufacturers). Further, we develop a model-based controller by inverting the plant to enable velocity tracking. The work is corroborated via simulations and experiments.

The SAE J1939 standard has been widely adopted by diesel engine manufacturers and serves as an update to the SAE J1708 and J1587 standards. Our research proposes to use to the estimated torque output from the engine, along with the engine speed and throttle pedal position information published to the CAN bus to procedurally generate a torque map of the engine. While some works in the past [2] have adopted a similar methodology to develop control algorithms for passenger cars using gasoline engines, controllers for diesel powered vehicles based on torque maps are relatively uncommon. To account for the high degree of non-linearity associated with diesel engines, some researchers have developed advanced models and controllers [3] for this purpose. We show that the non-linearity associated with a diesel power train can be reasonably accounted for using the aforementioned torque map approach.

First, a vehicle dynamics model was developed using Dymola. Parameter identification was performed on the model by collecting data from the vehicle CAN bus. Based on the tuned plant model a PID controller was implemented which utilized vehicle CAN information feedback on a real-time basis. All field experiments were conducted at Texas A&M's RELLIS campus, on an International Prostar (2012) tractor-semi trailer, supplied by Texas A&M Transportation Institute (TTI).

The thesis is divided as follows: Section 2 elucidates the model development work, Section 3 presents the parameter identification performed on the model. Controller development is presented in Section 4, followed by a discussion of the Results in Section 5 and a summary.

2. MODEL DEVELOPMENT

Dymola (Dynamic Modeling Laboratory) is a Modelica based tool for modelling and simulation widely used in a variety of automotive, aerospace and robotics applications [4]. This work utilizes the Vehicle Dynamics Library within Dymola, which contains various tools and templates for automotive components that can be adapted for a heavy-duty vehicle.

Overall Model

The overall truck model is presented below in Figure 1. All screenshots of the model presented in this work were taken from Dymola's Graphical User Interface and appropriately labelled. While not explicitly used in this work, the model incorporates features of the truck's environment for future work.

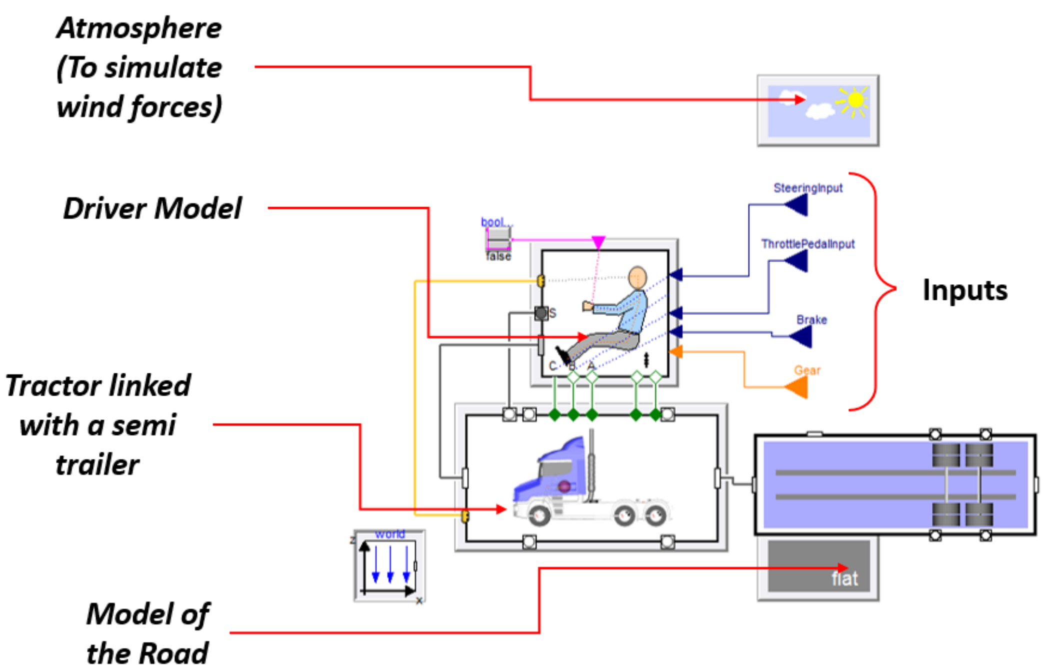


Figure 1: Overall Truck Model developed in Dymola

The atmosphere block allows specifying the wind vector, to account for aerodynamic drag. Gradient, traction and other road conditions can be specified as well. The external inputs to the model consist of driver actions i.e., steering wheel position, throttle pedal position, brake demand and gear selection. Given that the International Prostar truck supplied by TTI was an automated manual, an external gear shift scheduler was developed (not shown in above Figure). The tractor and trailer are modelled separately and are connected using a pin joint to account for the ‘fifth wheel’ hitch used to mount the semi-trailer. The model was designed to output the current wheel-based velocity of the truck, its current acceleration, engine speed and currently engaged gear ratio. It should be noted that these are the same outputs that will be utilized for real time feedback during field testing. Specific PGNs for these quantities are part of the SAE J1939 standard and can be purchased online [5].

Vehicle Subsystems

Tractor

The tractor model consists of an Engine, Transmission, Driveline and Cab body models. The driveline is connected to two axles in the rear, with twin hubs (two wheels on each side, per axle), and one steerable, single hub axle in the front. The axles are connected to the chassis of the tractor through an air suspension model, available as a template within Dymola. Front steerable tires are of type 295/75R22.5 while the rear axles have twin 295/80R22.5, matching the exact configuration in the truck. A schematic of the tractor model is shown below in Figure 2.

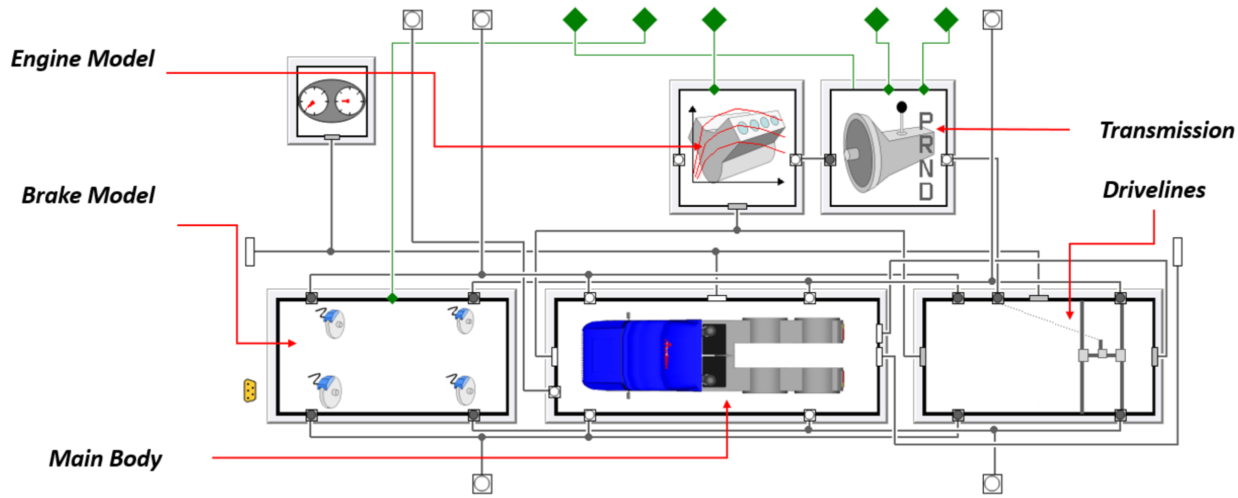


Figure 2: Dymola Model of the Tractor

The body of the cab, including the sleeper was assigned a net mass. Net frontal and lateral surface area of the cab were accounted for, to calculate the frontal and lateral aerodynamic drag forces. Additional details about the engine, driveline and brake models have also been presented below.

Engine Model

The core of the engine model consists of a Torque map. This is a 2D map with Throttle (%) and Engine Speed (RPM) as the independent variables and torque (Nm) as the output. Thus, the throttle pedal position input to the overall model is normalized and supplied to the engine model. A time lag of 1 second has been implemented in the throttle dynamics, to account for parasitic lags in the engine. Internal feedback of the speed of the camshaft provides the required engine speed value for the Torque map. The Torque map table was generated from experimental data and a description of the procedure has been presented in Section 3.

Transmission

Since the Eaton transmission installed in the truck was a 10-speed automated manual transmission, there was no need to implement a model of the torque converter. Instead, a linear clutch model was used. An external gear selection algorithm supplies the required gear number to the transmission model. A lookup table (generated via the CAN bus as described in Section 3) provides the gear ratio corresponding to each gear number. The transmission model utilizes these gear ratios to scale up/scale down the torque obtained from the engine. The output of the transmission axle is connected to the Driveline subsystem.

Driveline

The driveline consists of further geared assemblies representing the bevel gears in the truck and a final drive gear. The reduction ratio of the final drive gear was also obtained through experimental analysis.

Brakes

Heavy Duty vehicles typically use pneumatic brakes. For an example of a detailed model of such a pneumatic system, the reader can refer to [6]. Furthermore, the Prostar truck provided by TTI was configured with Bendix Wingman Fusion [7] system, which allowed for CAN based control. The Wingman system consisted of a low-level controller that utilizes a demanded deceleration value and appropriately controls the brake force to achieve the desired deceleration. The low-level controller automatically switches between engine braking (‘Jake brake’) at low braking demands and pneumatic system at higher demand. It was decided to model the brakes as friction-based disc brakes, akin to those commonly used in passenger cars, for the sake of simplicity. The validity of this assumption is revisited later in the document.

Semi-Trailer

The semi-trailer subsystem consists of a rigid container mounted to a chassis. Two axles on the rear also have twin hubs. All tires used are of type 295/80R22.5. Figure 3 below shows the trailer subsystem.

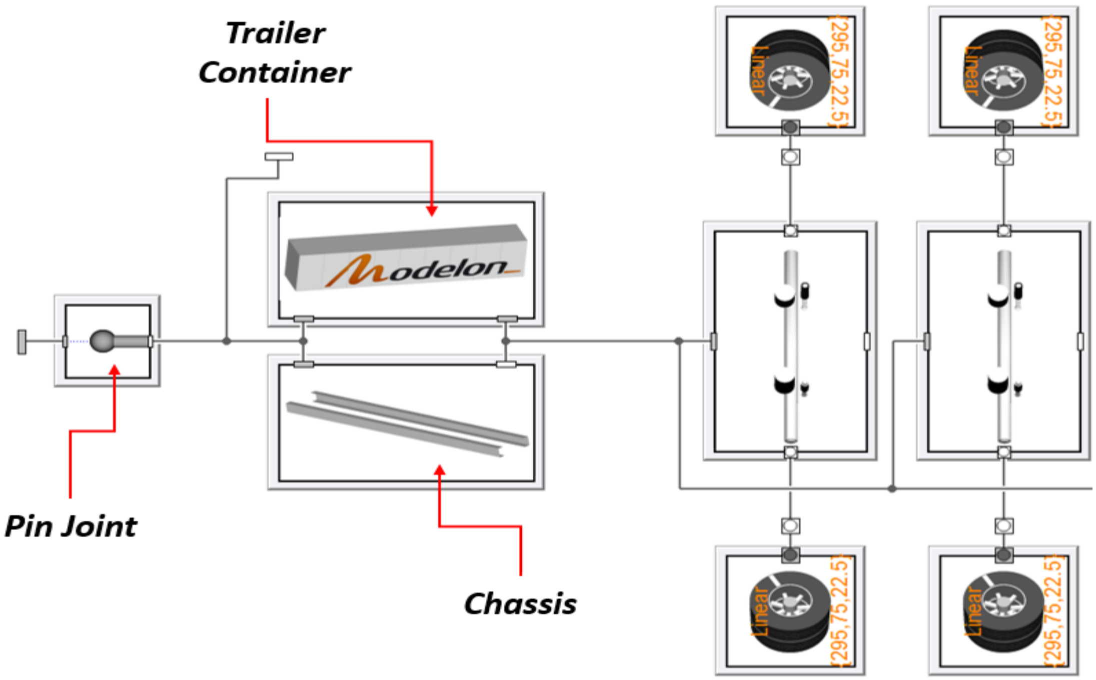


Figure 3: Dymola Model of the Semi-trailer

3. PARAMETER IDENTIFICATION

Once the model structure had been implemented in Dymola, it was necessary to tune parameters in the model to fit those of the Prostar truck. While some parameters were tuned by trial and error, most were calculated using data collected from carefully designed experiments. The parameter identification methodology has been broken down by subsystem and presented below:

Engine: Torque Map

The core of the engine model is the relation between the engine speed, throttle position and torque output. The SAE J1939 CAN specification at the vehicle application layer includes an estimate of the engine torque, normalized from 0 to 100. The Reference maximum torque output is also published as a separate signal. If the latter is not available, the maximum torque specification of the engine can typically be obtained from a brochure. Using these, the current (estimated) torque output of the engine can be recorded. This can then be combined with engine speed and throttle pedal position data to generate a 3D torque map with torque as the output.

Since the autonomy enabled vehicle available with TTI afforded the option to directly control the position of the throttle pedal through a linear actuator, the experiment designed was as follows: The truck was brought to a complete stop at the start of each run. A constant pedal height was selected and requested from the linear actuator. Then, the currently engaged gear number from the automated manual was monitored via the truck’s dashboard. The truck was allowed to move in a straight line and allowed to accelerate till the transmission shifted into the highest gear. The truck was then brought to a complete stop and this procedure was repeated for different throttle levels. A decommissioned runway at RELLIS campus was utilized for these runs. The throttle percentage, estimated torque from the engine, engine speed, current gear engaged, and its corresponding gear ratio were recorded from the CAN bus. The wheel-based velocity of the vehicle was also recorded, for use in transmission modeling, as described later in the document. Figure 4 shows the different throttle inputs that were run for the engine map tuning. The peaks in between the plateaus represent manual driving when the truck had to be turned around or moved into position between each run. Moreover, since the throttle control was mechanically enforced through a linear actuator pressing down on the throttle pedal instead of an electronic throttle control, the throttle pedal was subjected to vibrations arising due to the motion of the truck and wasn't perfectly constant. But these variations were relatively small and did not hinder the process of building a torque map.

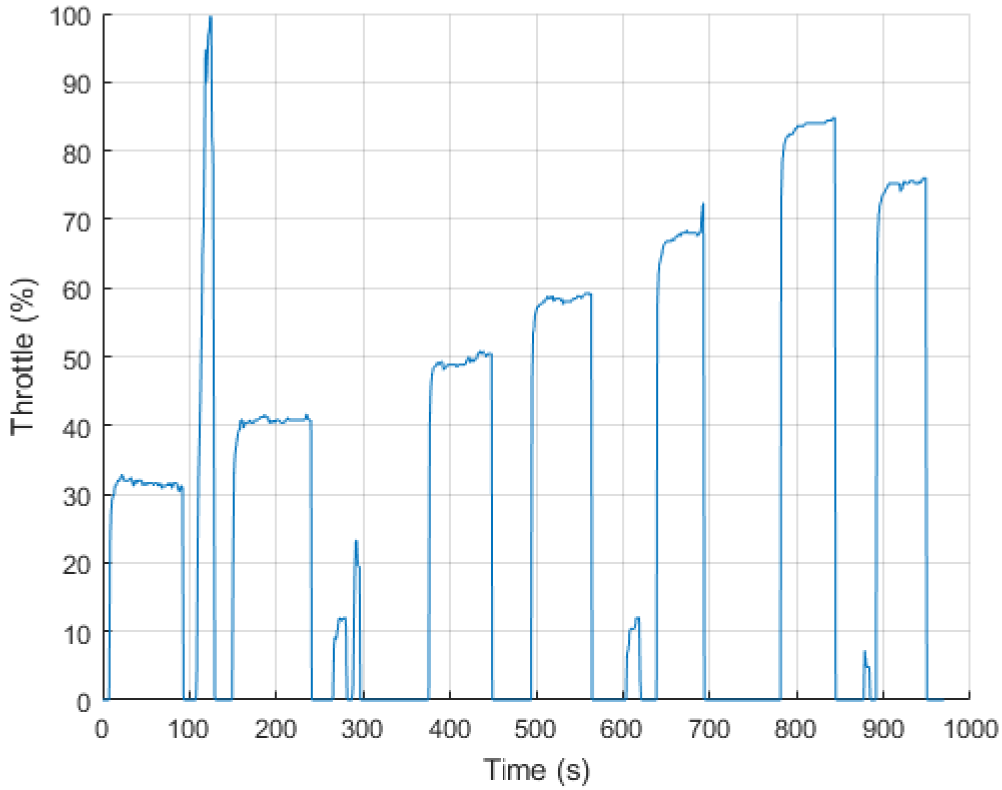


Figure 4: Throttle Level Recorded

Post processing the experimental data involved filtering out the portions of the data when the truck was manually accelerated by the driver and building a 3D point cloud from the Engine Speed, Throttle Percent and Engine Torque. A uniformly gridded surface was then fit from this point cloud using the 'gridfit' function, downloadable freely from the MathWorks Community File Exchange [8]. Moreover, it should be noted that engine braking torque is typically not published to the CAN bus. This means that negative torque output from the engine (at high engine speed and low throttle levels) cannot be measured directly. But, the gridfit function allows extrapolation of the surface, giving a reasonable estimate of the negative region of the torque map. *Figure* 5 shows the post processing performed on the engine map data collected and the corresponding surface that was fit to the empirical data points.

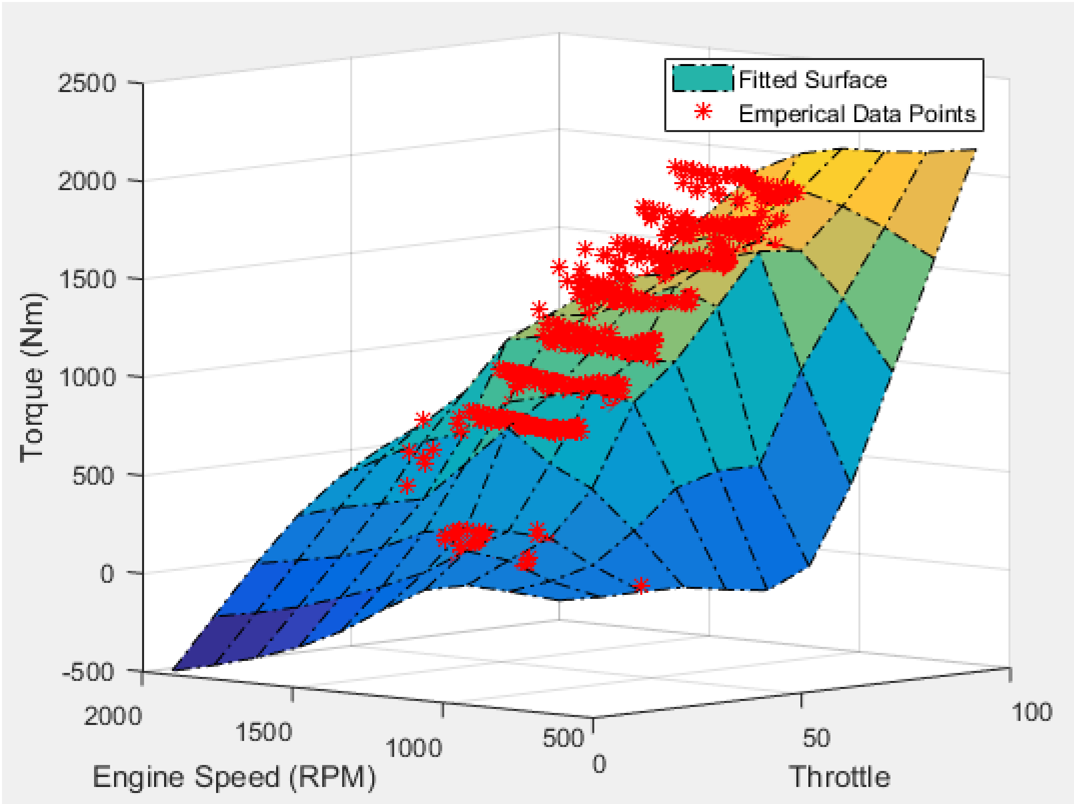


Figure 5: Engine Map - Data Points and Surface Fit

The surface fit can be represented as a 2D lookup table, which is the preferred format for use in the Dymola engine model. The lookup values have been reproduced below in Table 1.

Table 1: Engine Torque Output - 2D Lookup Map

Engine Speed (RPM)	Throttle Level (%)										
	0	10	20	30	40	50	60	70	80	90	100
614	86.7	77.0	79.0	74.9	36.7	6.8	103.2	492.2	1041.1	1593.5	2125.2
780	105.8	131.2	186.0	336.5	526.7	590.2	590.3	1051.2	1392.3	1724.9	2084.2
920	113.2	215.1	303.2	527.9	941.1	1125.2	1376.6	1633.7	1611.0	1811.7	2063.7
1060	67.7	240.7	313.8	634.6	974.2	1102.8	1457.9	1571.4	1641.2	1871.0	2057.7
1200	-58.2	107.3	305.4	794.3	927.9	1134.4	1364.1	1474.1	1439.0	1707.6	2009.0
1340	-191.8	13.8	266.1	588.8	873.8	1188.9	1264.9	1436.5	1366.9	1593.1	1903.3
1480	-293.3	-66.4	163.0	385.2	698.0	1143.3	1244.8	1339.9	965.9	1526.4	1693.6
1620	-369.5	-136.7	94.8	340.6	631.2	857.4	826.6	751.8	988.1	1184.2	1248.8
1760	-429.7	-188.4	52.9	295.9	521.8	672.6	712.7	579.0	709.1	803.8	831.4
1900	-481.9	-230.1	13.8	231.5	402.3	524.7	607.4	622.6	533.4	453.7	394.0

This table allows Dymola's engine block to calculate the torque output based on the current engine speed and throttle demand from external input (driver/controller).

Transmission: Gear Ratio Table

The Electronic Transmission Controller typically publishes the current gear engaged as well as the corresponding gear ratio as sperate signals to the vehicle CAN bus. Thus, by cycling through the gears, a gear table can be generated, as shown in Table 2.

The last two entries are for the two reverse gears available in the truck. For the purpose of this work, we only deal with forward gears during longitudinal control.

Table 2: Gear Ratio Table for Transmission

Gear Number	Gear Ratio
1	12.8
2	9.251
3	6.761
4	4.901
5	3.579
6	2.611
7	1.888
8	1.38
9	1
10	0.73
R1	-13.63
R2	-2.78

Driveline: Final Drive Gear

After accounting for the gear ratios within the automated-manual transmission system on the truck, it was observed that the velocity of the simulated model in Dymola did not match that of the truck even on matching engine speeds. Thus, it was concluded that a driveline gear, external to the transmission, must be present in the truck, as is common in many heavy-duty vehicles.

Unlike the transmission system, the specifications of this gear, located between the transmission and the wheels is not published to the CAN bus. But it can be calculated from available information. The engine speed (in RPM) and wheel-based speed of the vehicle are both available on the standard J1939 CAN bus. Once the transmission gear ratios have been determined, the speed of the transmission axle can be calculated by dividing the engine speed with the gear ratio of the currently engaged transmission gear. Dividing the transmission output speed by the wheel speed (after adjusting for units) will provide the value of the final drive gear ratio. That is,

$$Final\ Drive\ Ratio = \frac{Engine\ Speed\ (RPM)}{Transmission\ Gear\ Ratio * Wheel\ Speed\ (RPM)}$$

This analysis was applied to the data collected earlier. A plot of the final drive ratio over a portion of the run is shown in Figure 6. The peaks in the graph coincide with gear shifting, during which the transmission is not locked up with the engine and the relation above does not hold. Thus, ignoring the peaks, we observe a steady ratio. Consequently, the final drive ratio was identified to be 3.39, for use in Dymola. The identified value is also shown in the figure.

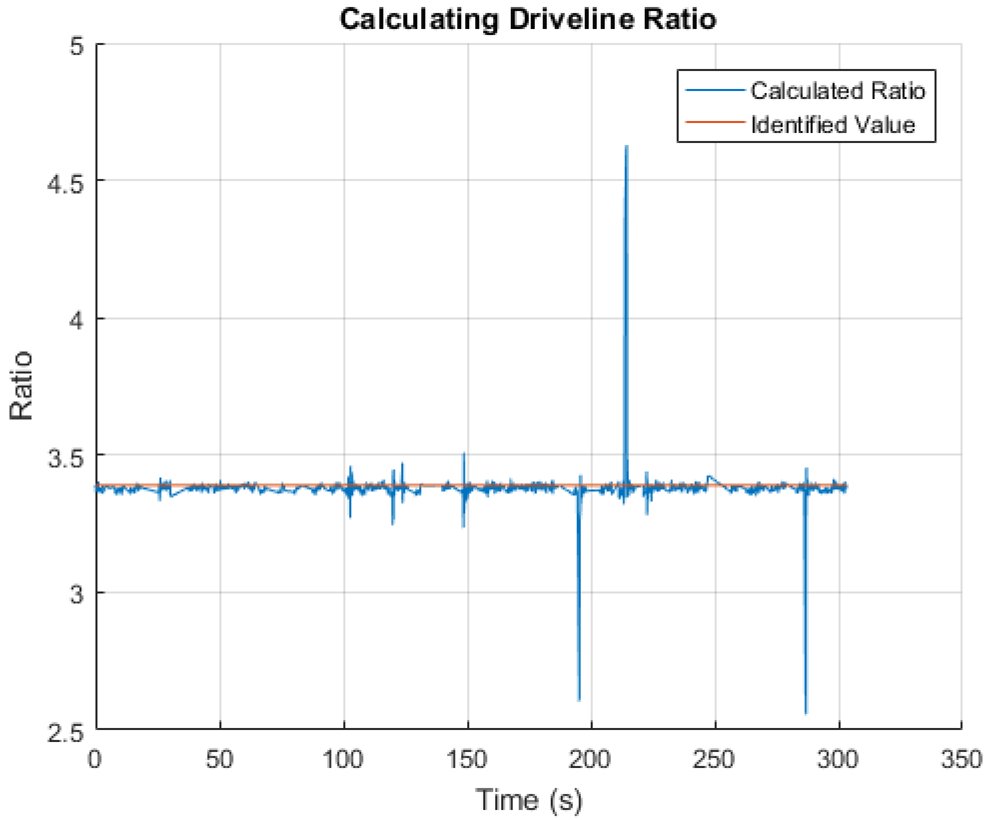


Figure 6: Final Drive Ratio Identification

Brake System

As mentioned above, the truck's brake system was modelled using disc brakes instead of developing a detailed pneumatic system. It was observed that this was sufficient to faithfully reproduce the performance of the CAN – based Bendix braking system installed on the truck. For the purpose of this project, the maximum braking deceleration achievable from the Dymola model was assumed to be 8 m/s^2. Given the combined mass of the tractor and trailer to be approximately 24,000 kg (no cargo in trailer), the total braking force required for maximum deceleration can be calculated and then distributed among the five axles in the vehicle. Thus, each wheel hub (two wheel hubs per axle) was assigned two clamping pads, with friction coefficient 0.3. Effective radius of the discs was taken to be 15 cm. The hydraulic booster system's specifications were set accordingly, to achieve the required braking pressure on the brake clamps. We have assumed that the brake force is evenly distributed amongst all the vehicle's axles. Consequently the model's performance is expected to change if the trailer is disengaged from the tractor.

It should be noted that in this model development process, a few of the components were used from Dymola's library without any modification. For example, the air suspension systems for the axles in the tractor and trailer were used as is, since it was determined that further tuning wouldn't be necessary for improvements in longitudinal dynamics. For the chassis as well, the default configuration was maintained with only the mass of the chassis and body adjusted to the estimated values of the Prostar Truck.

Finally, for wind drag forces, the frontal aerodynamic drag coefficient of the tractor was set to 0.6 with a frontal area of $8m^2$. The side drag coefficient was set to 0.5 with side area of the tractor body as 20 m^2. These values were 'order of magnitude' approximations taken from rough estimates of the dimensions of the truck. Other aspects, such as rolling resistance, and road gradients etc. were set to zero. If higher fidelity is required from the model, these can be experimentally determined as well.

After performing parameter identification on the model, the model's performance was evaluation in comparison with the truck. Dymola provides a Simulink toolbox that, with appropriate licenses, allows connecting a Simulink file to Dymola. The simulation was first performed in open loop. That is, the truck was manually driven in the field where inputs to the truck (throttle pedal position, brake demand, and current gear) were recorded and the same were supplied to the model as a time series. The position of the simulated model was found to match reasonably well with the recorded trajectory information. The comparison is shown in Figure 7.

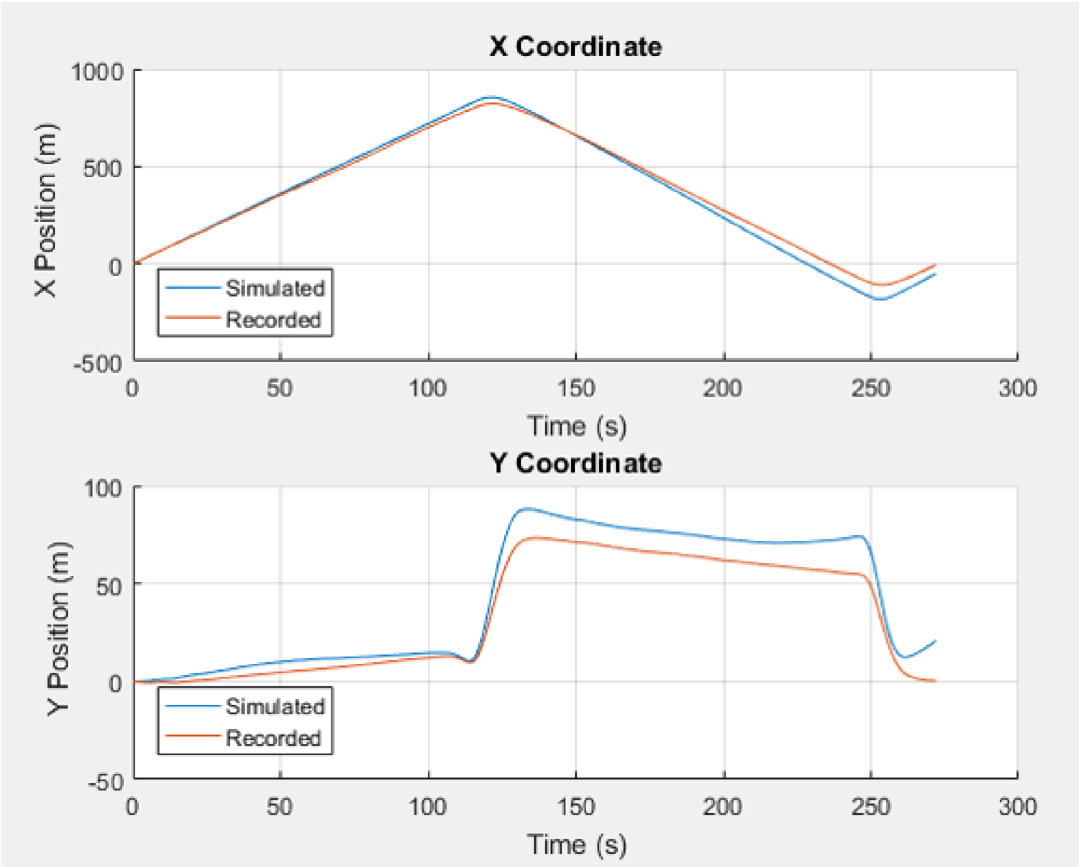


Figure 7: GPS (UTM) Comparison of Simulated vs. Recorded Trajectories in Open Loop

4. CONTROLLER DEVELOPMENT

Once the model identification was completed, a model based longitudinal controller was developed. The controller was first implemented in closed-loop with the Dymola plant model in Simulink and tuned to achieve satisfactory velocity tracking performance. Then, the same controller was also implemented on the MicroAutobox platform in the truck. This section provides description of the controller and the closed loop simulation setup. A comparison of results from the simulation and field testing is provided in the following section.

Controller Overview

Simulink was chosen as the controller development platform given the constraints of the hardware installed in the Truck provided by TTI. The MicroAutobox hardware provided by dSPACE requires the use of RTI toolbox in Simulink for algorithm development. But, the controller developed here is not platform specific and can be easily implemented in other environments. *Figure* 8 below provides an overview of the controller developed in Simulink.

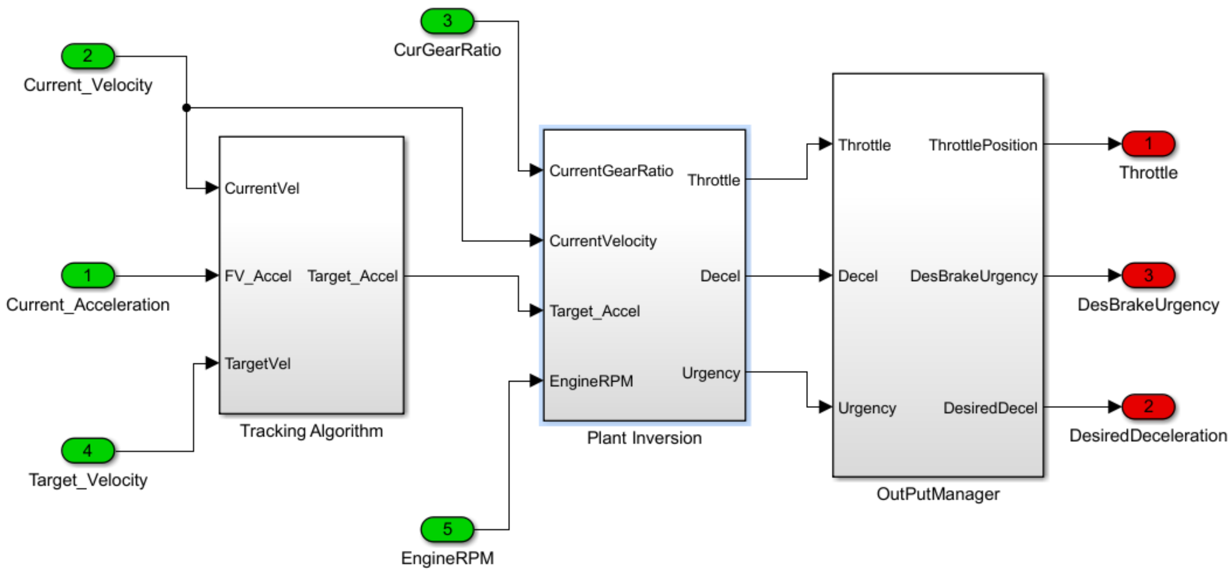


Figure 8: Controller Overview

The controller requires the current velocity of the vehicle and its current acceleration as feedback terms from the plant. These are supplied by the CAN bus in the case of field testing or supplied as outputs from the Dymola model in case of desktop simulation. A target velocity is also supplied as an input. The controller outputs the required throttle pedal position, and braking commands to the brake controller in the plant. The brake controller requires an ‘Urgency’ scalar (0 -100, no unit) and a deceleration (in m/s^2) value. These controller outputs are fed to the Dymola model, or in the case of the truck, fed to the linear actuator mounted to the throttle pedal, and the Bendix low level brake controller via the J1939 CAN bus.

Controller: Tracking Algorithm

The Tracking Algorithm contains PID logic to generate a required acceleration demand based on the current velocity, current acceleration and target velocity. The errors are calculated using a discrete derivative and discrete integrator block for the I and D terms. In addition, the integral term is reset to zero as the current velocity approaches within 1 m/s of the target velocity. That is, it effectively transitions to a PD controller for small tracking errors. The PID gains shown in Figure 9 were obtained after tuning.

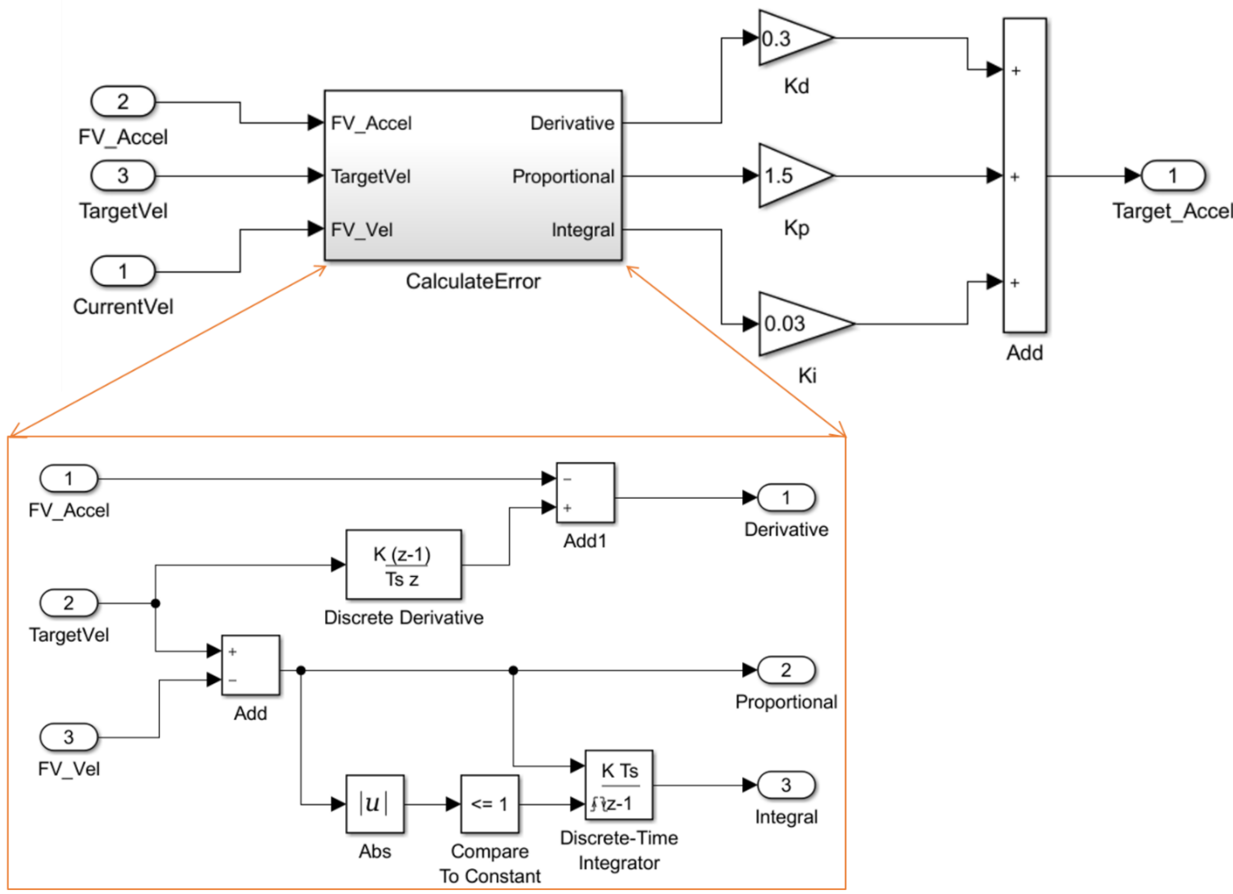


Figure 9: Tracking Algorithm

Controller: Plant Inversion

Once the desired acceleration is calculated from the PID logic, appropriate throttle / brake demands need to be obtained. This is the model based design aspect of the controller since we utilize information gleaned through parameter identification earlier in the process to generate the throttle demand. The Plant Inversion logic is presented in Figure 10.

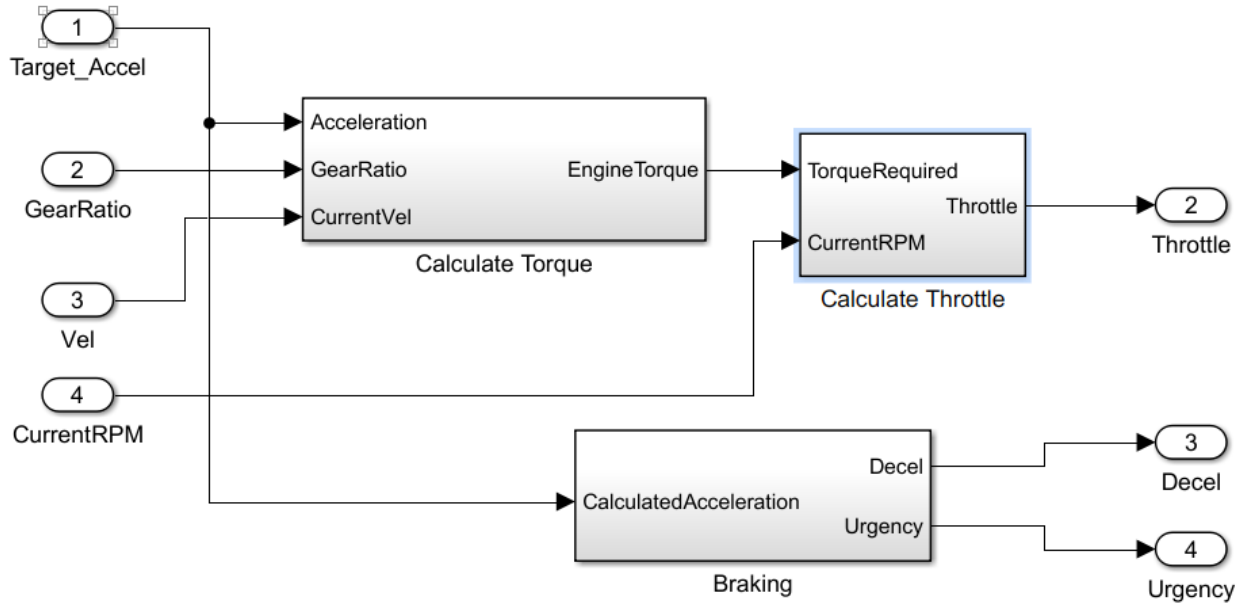


Figure 10: Plant Inversion

The algorithm is split into two branches depending on whether the acceleration desired is positive or negative. For negative desired acceleration (braking) there is no further processing required as the Bendix brake controller and the Dymola plant model are both designed to accept a deceleration value in m/s^2 as inputs. For a positive target acceleration, first the torque required from the engine is calculated, as shown in Figure 11. This involves calculating the torque at the wheel and then dividing by the final drive ratio and the gear ratio of the currently engaged gear to obtain the Engine Torque required.

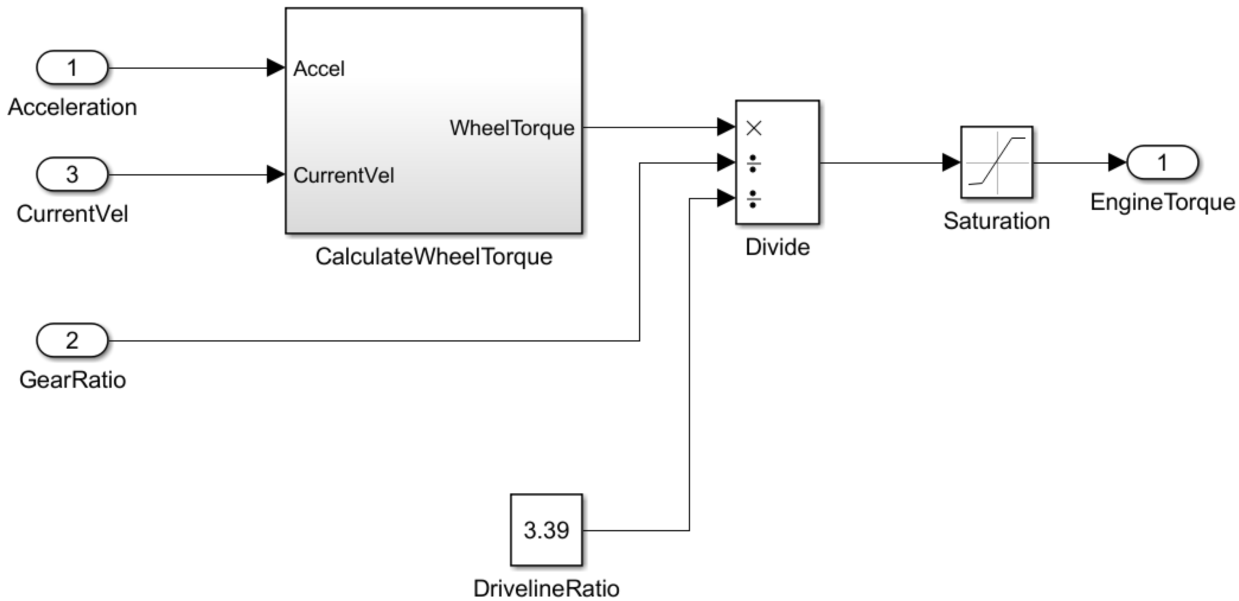


Figure 11: Torque Demand Calculation

A Saturation block has been added with the upper saturation limit set to 2305 Nm, which is the maximum torque output possible from the engine. This information is typically published to the vehicle CAN; otherwise it can be obtained from openly published brochures from the engine manufacturer. The Wheel torque calculation (not shown) involves summing the forces from moving the vehicle (F=mass*acceleration) with the aerodynamic drag force and an estimated rolling resistance. This sum represents the net load of the vehicle at the tires. Given the radius of the tires (in this case 0.538m), this can be converted to a Wheel Torque load.

Finally, the torque demand from the engine is converted to a required throttle value by inverting the Engine Torque map that was developed earlier during model identification. While the map developed earlier in Table 1 provides the torque output from the engine given the throttle and current engine speed, the inversion of the table allows interpolating the value of the required throttle given the engine speed and torque demand. The logic is presented visually in Figure 12.

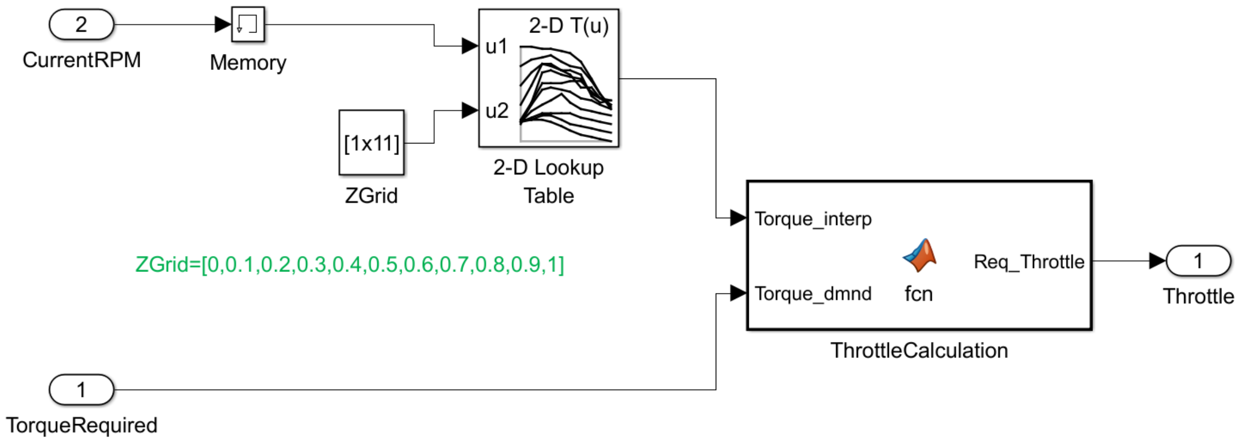


Figure 12: Calculating Throttle from Torque

Due to the absence of a Simulink Coder enabled 2D interpolation algorithm function in MATLAB {interp2() cannot be built into an application for use in Real Time systems such as the MicroAutobox}, a workaround was developed. First, a lookup table is used to extract an array of torque outputs for the current engine speed. Each element in this array corresponds to a specific throttle level. One dimensional interpolation is then performed on this array to calculate the required throttle value. The script for the MATLAB function in Figure 12 is presented in full in the Appendix. The unit delay ('Memory') block is present to resolve an algebraic loop in Simulink.

Finally, the Output Manager block in *Figure* 8 contains some fail-safe logic for practical implementation. For example, a lower and upper limit (0 -100) is implemented for the throttle. The throttle command is also disabled (stubbed to zero) during braking action.

Closed Loop Implementation

The developed controller was implemented first in closed loop with the Dymola plant model to tune the PID gains. The reference velocity can be supplied either manually or as a time series from the workspace. This was later used to compare the performance of the closed loop simulation with that of the truck by supplying the same target velocity profile to both systems. An overview of the closed loop simulation is presented in Figure 13. Signal Lines have been color coded in an attempt to improve readability.

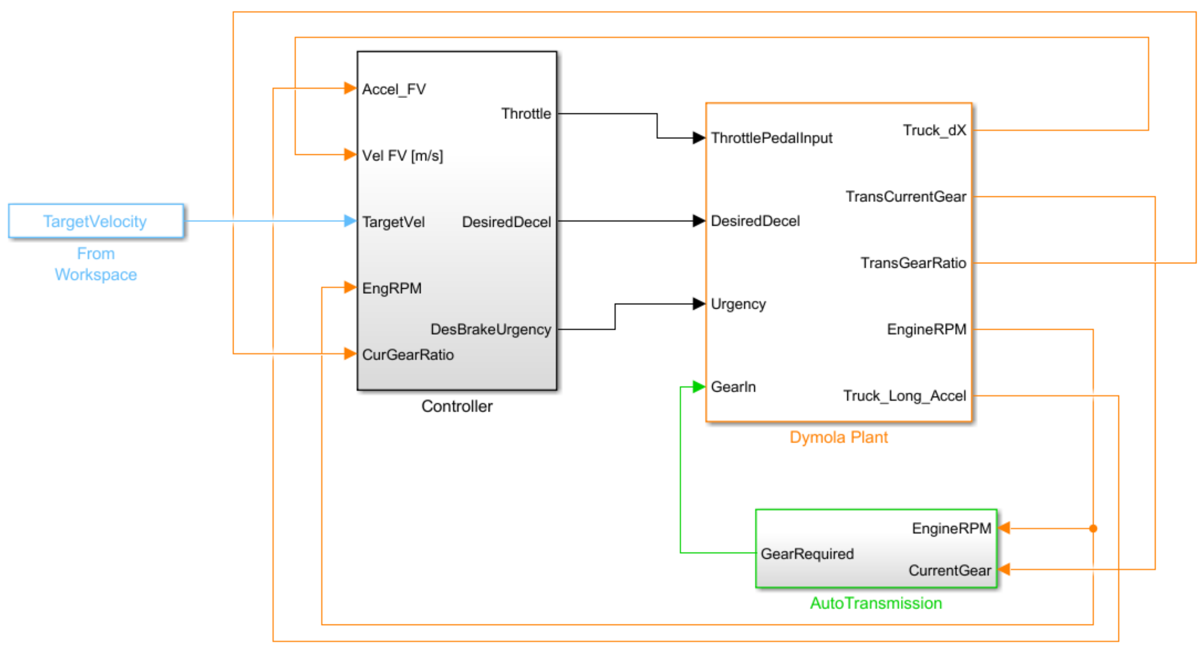


Figure 13: Closed Loop Simulation in Simulink

The tuned PID gains from desktop based closed loop simulation are encapsulated in Table 3.

Table 3: Tuned PID Gains from Desktop Simulations

Gain	Tuned Value
Proportional Gain (K_P)	1.9
Derivative Gain (K_D)	0.3
Integral Gain (K_I)	0.03

As mentioned earlier, the Dymola plant model functions as a truck with manual transmission. That is, the gear number needs to be supplied to the plant model as in input. But, the truck used for field testing is an ‘automated-manual’ transmission, which consists of a transmission controller with a gear shifting logic. Typically, gear shift tables are proprietary information. But, based on cumulative observations of the behavior of the truck in experiments, a shift scheduler was developed (Labelled ‘Auto Transmission’ in Figure 13). In reality, the shifting logic would be based on the estimated torque load on the engine and possibly other factors. Moreover, the transmission controller in the truck also allows for ‘skipping’ a gear under high acceleration demands. For the purposes of this work, these behaviors are ignored. Instead, the algorithm shifts up by one gear whenever the engine speed increases beyond 1450 RPM and shifts down when the speed drops below 950 RPM. A schematic with details of this logic implemented in Simulink is presented in the Appendix in Figure 18.

5. RESULTS AND DISCUSSION

Since the truck provided by TTI is equipped with a dSPACE microAutobox, the controller developed in Simulink could be implemented directly to the real-time hardware. The experimental run consisted of supplying a target velocity to the controller. A rate limiter, with upper and lower caps of ± 2 m/s^2 was also implemented to avoid instantaneous changes in velocities.

Given that PID tuning was already performed on using closed loop simulation with Dymola, very minimal tuning had to be performed in field. The final, tuned PID gains are presented in Table 4. On comparing with Table 3, the value of developing a high-fidelity plant model and running simulations before performing experiments with real vehicles is apparent.

Table 4: Tuned PID Gains from Field Experiments

Gain	**Tuned Value**
Proportional Gain (K_P)	1.5
Derivative Gain (K_D)	0.3
Integral Gain (K_I)	0.03

The wheel-based velocity, along with the target velocity was recorded over the course of the experiment. Other quantities, such as the throttle and brake commands issued by the controller, acceleration of the vehicle and the gear number engaged (from J1939 CAN bus) were also recorded for further post processing.

The same target velocity profile was also supplied to the Dymola plant model in a closed loop simulation setup. The PID gains were changed in the simulation to match that of the field controller. After running the simulation, the velocity generated by the model was recorded over the same time span. The recorded wheel-based velocity of the truck in an experimental run, the model's predicted velocity and commanded target profile are shown below in Figure 14. The complete run lasted approximately 10 minutes.

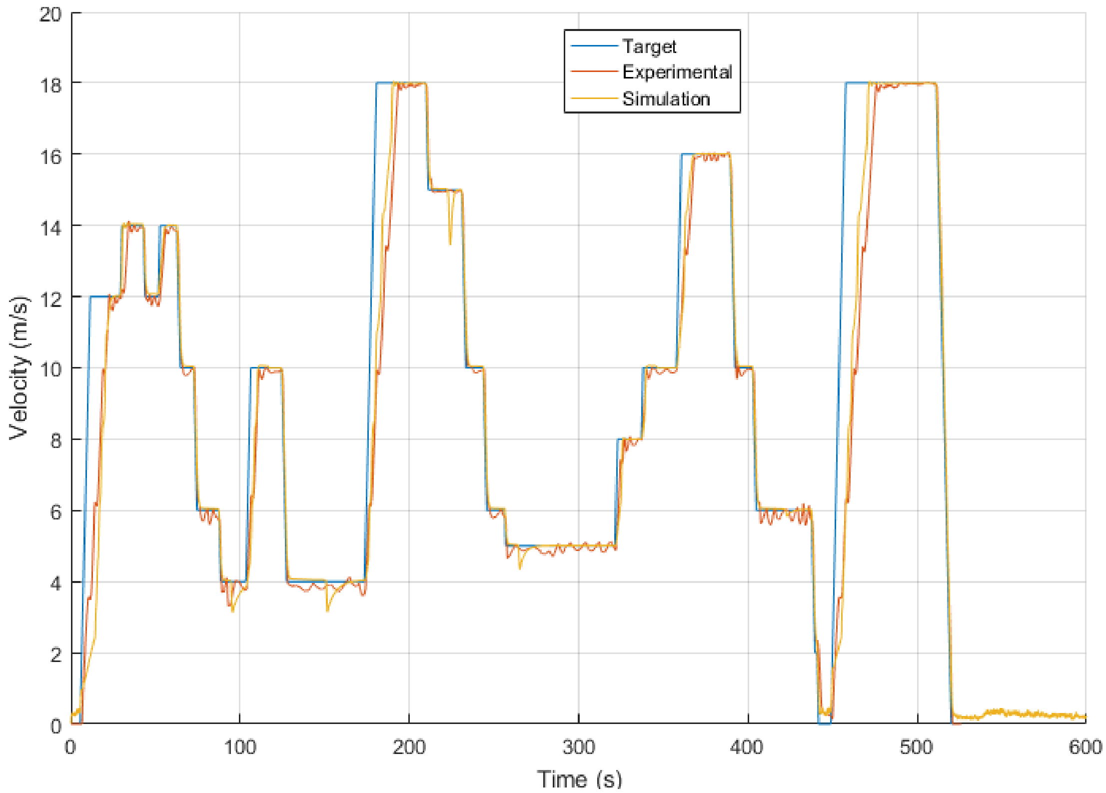


Figure 14: Comparison of Target, Simulated and Experimental Velocity profiles

A few observations can be drawn from these results:

1. The Truck (in field experiments) tracks the commanded velocity well except during severe acceleration demands. When the reference velocity is constant, both the experimental and simulated trucks stay within 0.5 m/s (~ 1MPH) of the reference. It should be noted that even though the recorded velocity appears to oscillate when the target velocity is constant (for example, between T =250s to T =320s), these oscillations

are low in amplitude and frequency, so they are not perceivable by riders within the truck. In fact, no acceleration or deceleration was picked up by truck's onboard IMU (based on the data published to the CAN bus over this time span). A rescaled figure of the velocity profiles is provided in Figure 15 , for perspective. It can be observed that the amplitude of the oscillations are typically less than 0.2 m/s, with a time period of 6 -10 seconds.

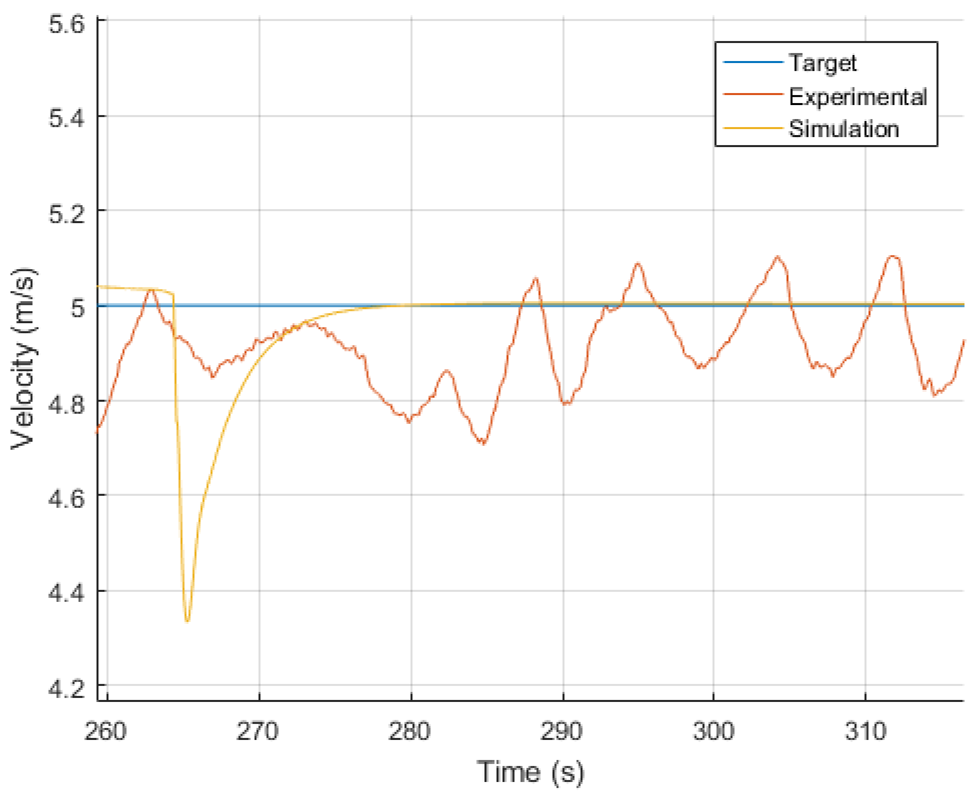


Figure 15: A Rescaled portion of the Velocity Profiles

2. The controller performance is not expressly affected by the magnitude of the target velocity supplied. Given that the experimental run covered a significant range of speeds (0- 40 mph) and that there was no need to re-tune the controller to achieve stable

performance at different speeds, it can be reasonably postulated that the controller will track a constant velocity profile at all speeds attainable by the vehicle. High speed field testing on the truck was limited by the availability of space on RELLIS campus.

3. The controller performs relatively better in braking action than during acceleration. Moreover, the braking profiles of the field-testing match near-perfectly with the simulated profile. Thus, we can conclude that the assumptions made during brake modeling were reasonable.
4. The Dymola plant model, running in closed loop with the controller manages to closely predict the velocity profile obtained from field experiments, highlighting the utility of developing a detailed plant model. In fact, Figure 16 shows the tracking error (error between target and achieved velocities) for the simulation and experiment.

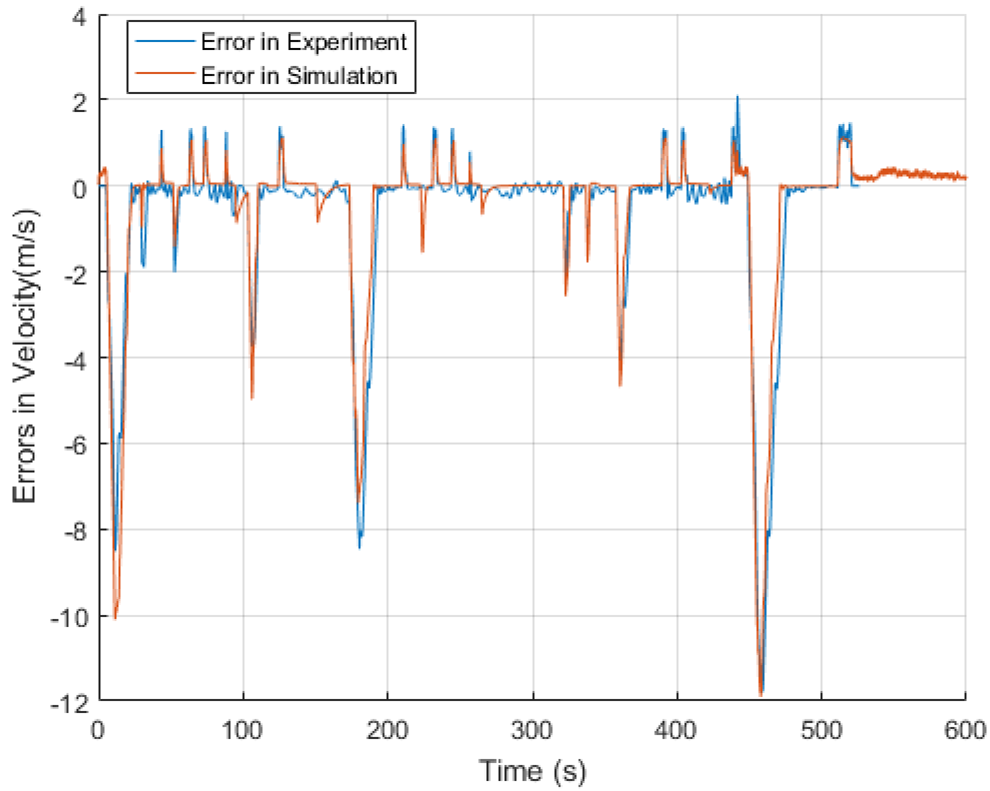
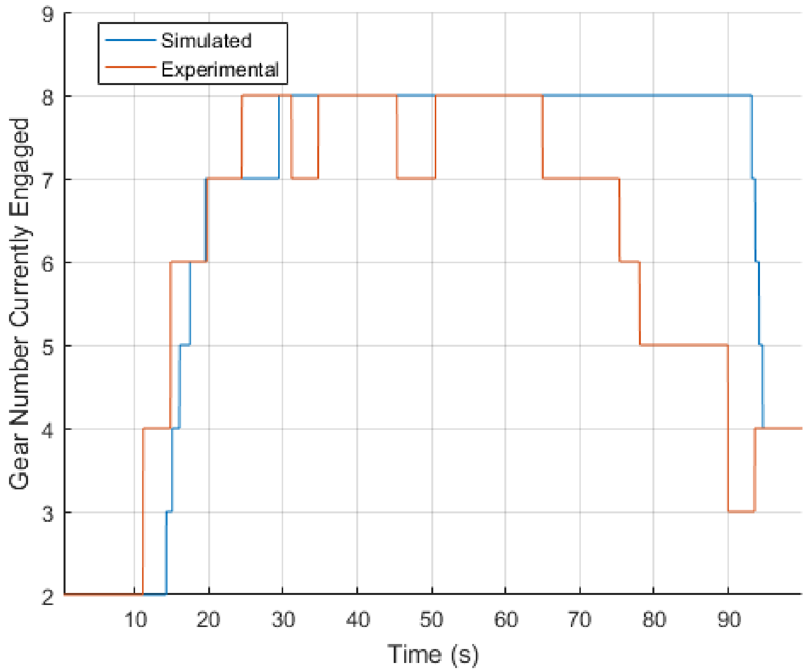


Figure 16: Velocity Tracking Errors – Simulation vs. Experiment

Some of the observed drawbacks of the developed controller are also addressed below:

1. It can be noted from Figure 14, that in both the simulated and experimental cases, the truck is unable to keep up with demanded velocity during the ramp-up portions. More than the performance of the controller, this is due to the limitations of the maximum acceleration attainable by the diesel engine of the truck. It is worthwhile to note that the throttle pedal command output from the controller was at 100% during the ramp-up portions. The ramp has a slope of +2 m/s^2. The semi-trailer was observed to achieve a maximum acceleration of 1.3 m/s^2 with wide open throttle in a separate experiment.

2. When the commanded velocity is held at 0, the truck comes to a complete stop in experiment, but the simulated model fails to do so. There is a small residual error (~0.3 m/s) in the velocity of the model, which does not appear to diminish. This indicates imperfections in the braking subsystem of the Dymola model, which appears to be unresponsive at very low velocities. Apart from improving the robustness of the brake system, this could possibly also be mitigated by an implementing an intelligent clutch in the model which disengages the engine from the transmission.
3. While the overall performance of the model in simulating the velocity profile of the truck is acceptable, the simulation is closer to experiment during braking and constant velocity tracking than during acceleration. On further investigation it was observed that the shift scheduling algorithm developed in Simulink was the leading cause of this discrepancy. As noted earlier, the transmission controller in the truck often ‘skips’ a gear during upshift/downshift at high throttle/brake demands. Due to lack of detailed information about shift scheduling, this logic is absent in the model. Consequently, the simulated and real truck were always not running on the same transmission gear. This is apparent from directly comparing the gear profiles of the simulation and experiment. A representative portion, spanning 100 seconds, is shown in Figure 17. We can reasonably conclude that an improved gear selector model for desktop-based simulation would help alleviate the discrepancy.



a

Figure 17: Comparing Shift Scheduling – Simulation vs. Experiment

Thus, it can be concluded that the model-based longitudinal controller developed performs satisfactorily for velocity tracking. Further, the fidelity of the Dymola model to the actual vehicle was also established. Its utility in developing control algorithms is evident from the fact that once the controller was tuned in simulation, very little tuning had to be done in the field to achieve required performance. Future work would include further improvements of the Dymola model, specifically in the transmission and brake subsystems. Other controller policies could also be explored, such as constant spacing or constant time headway algorithms for implementation in a platoon for vehicles.

6. SUMMARY

This work presents a model construction and parameter identification process of the longitudinal dynamics of a heavy-duty vehicle, for implementing a velocity-tracking algorithm. The main contribution of this work is that a usable plant model was developed using the J1939 CAN bus signals, without the need for any manufacturer privileged information of the truck. Templates available in Dymola were adapted as necessary for model construction. Once the model parameters were tuned, a PID controller was developed and tested in closed loop simulations before implementation in field tests. The results showed that the velocity control algorithm performed satisfactorily in both cases. Then, the closed loop performance of the model was compared with that of the empirical performance of the truck. From this comparison, the viability of the developed model was confirmed. Potential areas of improvement in the model have also been identified for future work.

REFERENCES

[1] Xiao-Yun Lu & J. Karl Hedrick, 2005. Heavy-duty vehicle modelling and longitudinal control. *Vehicle System Dynamics*, Volume 43, Issue 9, Pages 653-669. DOI: 10.1080/00423110412331282931

[2] Attia, Rachid & Orjuela, Rodolfo & Basset, Michel, 2012. Longitudinal Control for Automated Vehicle Guidance. *IFAC Proceedings and Volumes*, Volume 45, Issue 30. Pages 65-71. DOI: 10.3182/20121023-3-FR-4025.00049.

[3] Kolmanovsky, I., P. Moraal, M. van Nieuwstadt and A.G. Stefanpolous,1999. Issues in modeling and control of variable geometry turbocharged engines. *Systems Modeling and Optimization*, Hall/CRC Research Notes in Mathematics, Pages 436-445.

[4] 'Dymola Systems Engineering'. *Dassault Systems*. URL: https://www.3ds.com/products-services/catia/products/dymola/ Accessed 4 September 2018.

[5] 'Vehicle Application Layer.' Surface Vehicle Recommended Practice, J1939 -71. *SAE International*. URL: saemobilus.sae.org/content/J1939/71_201309. Accessed 22 October 2018.

[6] Subramanian, Shankar & Darbha, Swaroop & Rajagopal, K.R., 2003. Modeling the Pneumatic Subsystem of an S-cam Air Brake System. *Journal of Dynamic Systems, Measurement, and Control*. Vol 126, Issue 1. Pages 36 -46. DOI: 10.1115/1.1666893.

[7] 'Bendix® Wingman® Fusion™'. *Bendix Commercial Vehicle Systems*. URL: www.bendix.com/en/products/wingman_fusion/standard_page_4.jsp. Accessed 4 September 2018

[8] John D’Errico, 2016. Surface Fitting using gridfit. *Mathworks* URL:

https://www.mathworks.com/matlabcentral/fileexchange/8998-surface-fitting-using-gridfit.

Accessed: 22 October 2018.

APPENDIX A

MATLAB CODE FOR THROTTLE CALCULATION

Please refer to Figure 12 for the relevant subsystem.

```
function Req_Throttle = fcn(Torque_interp,Torque_dmnd)
Req_Throttle=double(0);
Thr_table=[0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1];
%manual interp function
for idx=1:(length(Torque_interp)-1)

    lower=Torque_interp(idx);
    upper=Torque_interp(idx+1);
    if (Torque_dmnd<upper)&&(Torque_dmnd>=lower)

      % In range, interpolate linearly
       Req_Throttle=0.1*((Torque_dmnd-lower)/(upper-lower))+Thr_table(idx);
       idx=(length(Torque_interp)-1);
       break;% quit for loop

else
       idx=idx+1;
   end;
end;

MaxTorque=double(0);
MaxTorque=double(max(Torque_interp,[],1));
MinTorque=double(0);
MinTorque=double(min(Torque_interp,[],1));
if Torque_dmnd >= MaxTorque
   Req_Throttle=1;
elseif Torque_dmnd <= MinTorque
   Req_Throttle=0;
end
end
```

APPENDIX B

GEAR SHIFT LOGIC FOR CLOSED LOOP SIMULATION

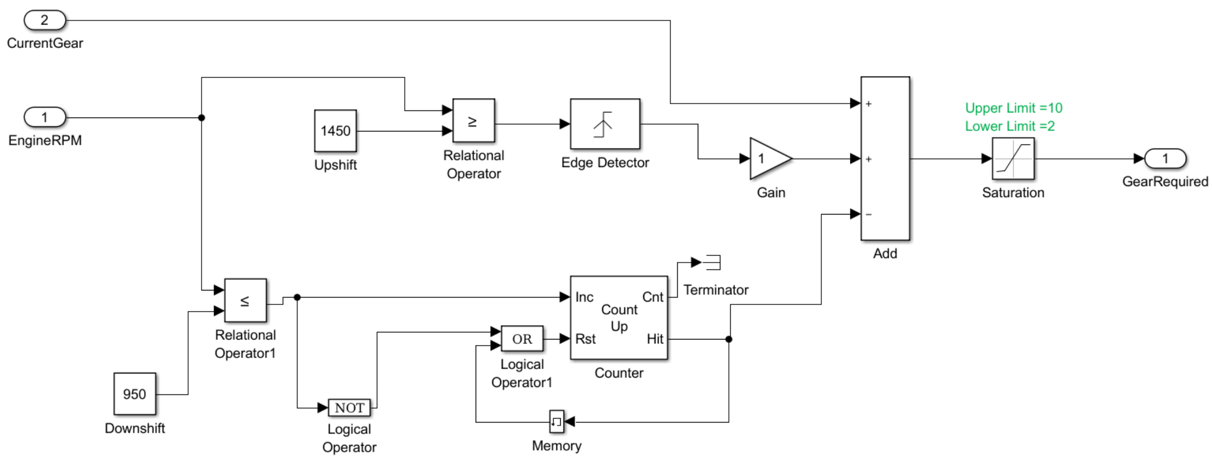


Figure 18: Logic for Gear Shifter